DOCUMENT RESUME

ED 310 744 IR 013 942

AUTHOR Evans, Richard M.; Reed, Robert W.
TITLE Relating Disparate Computer Training Task Data Files.
PUB DATE 28 Mar 89
NOTE 7p.; Paper presented at the Annual Meeting of the American Educational Research Association (San Francisco, CA, March 25-30, 1989).
PUB TYPE Guides - Non-Classroom Use (055) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; *Computer Software; *Information Retrieval; Military Training; Postsecondary Education
IDENTIFIERS American Standard Code Information Interchange; *Compatibility (Information)

ABSTRACT

Military training information is not available from a single, well-structured electronic source. It typically exists in multiple formats created by multiple software products, usually incompatible. The training analyst thus has the job of collecting that which is available and converting it to a format that can be processed by existing software. Further complicating the job is language ambiguity. The use of different terms or phrases when referring to the same topic is an inevitable outcome of many authors working on many files. Methods for converting information on training tasks stored in multiple incompatible records into a common form--American Standard Code for Information Interchange (ASCII)--and for working through the language ambiguities are presented. An example of the procedure is provided, and possible future applications of the procedure are described. (GL)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

RELATING DISPARATE COMPUTER TRAINING TASK DATA FILES[1] [2]

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Richard M. Evans
and
Robert W. Reed

Defense Training and Performance Data Center (TPDC)
3280 Progress Drive, Orlando, Florida 32826

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Richard M.Evans

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ED310744

Training information is not available from a single, well structured, electronic source. When found in this form, it typically exists in multiple formats created by multiple software products, usually incompatible. The training analyst thus has the job of collecting that which is available and converting it to a format which can be processed by existing software.

Further complicating the the job is language ambiguity. The use of different terms or phrases when referring to the same topic is an inevitable outcome of many authors working on many files. The problem of analyzing training information is then four problems:

- o Finding sources of information to study
- o Converting to a single electronic form
- o Reducing the volume by organizing into key terms
- o Working through the use/misuse of the language

Objective

This paper will not address the first problem--where to find sources of information. It will make a suggestion concerning the form into which the information should be converted. The discussion will concentrate on methods for working through the language ambiguities in order to provide the analyst a baseline from which to begin work. The actual relating takes place in the mind of the training professional who can understand the meaning of the records when juxtaposed by our procedure.

Background

Tasks are the least common denominator for describing the unit and individual performance in the military training system. Tasks link the various programs of instruction in formal military schools with occupations, manpower and personnel, warfighting equipment, training equipment.

We have been working on the problem (or four problems, as defined above) since September, 1987, when prototype programs were written for research purposes using the APL*PLUS Application Development System. There was little available to determine the essentials of a "task" at that time--only that the long verbal descriptions presented special problems for a data base management system. Data elements fit nicely into a dBase or FOCUS data base management system, but what is a task data element?

Actually, "task" turned out to have many data elements, including number, title (with verb, object, and qualifier), conditions under which the task is observed, and a standard for successful demonstration of the task. Each service component also had additional elements they presented with each task, such as training steps, administrative instructions, and references.

So a task statement does not stand on its own. There are perhaps 15-20 data elements that elaborate the task. Each element must really be considered in the context of the whole task. While "task" is more than the sum of its parts, the "good Gestalt" is necessary for its use in a military training environment. Any attempt to relate data elements across tasks is fraught with error possibilities. And, even this "least common denominator" can be broken down into its elements; e.g., title, conditions, standard, etc. While these "data elements" can be further divided into words, characters, and numbers, meaning is almost completely lost when you do. A task data element can not meaningfully be further divided.

All of this gives problems to a traditional data base management system. Should the complete task be stored in one record? If so, is there a file structure being imposed on something that is essentially a "flat file?" If so, the data base system is performing as a wasteful word processor. Should the task data elements be stored as records linked to the task? Then there may be needless complication in retrieving all of the tasks supporting some 3000 military occupations across four service components. TPDC sought other options.

---

1. A paper presented to the Military Education and Training Special Interest Group, American Educational Research Association Annual Meeting, Session 19.35, San Francisco, California, Tuesday, 28 March 1989.

2. The views in this paper are those of the authors, and not necessarily those of the U. S. Department of Defense.

IR013942

BEST COPY AVAILABLE

Method

A method for finding unrelated information in disparate text files is described by these steps:

1. Convert to ASCII

   a. The ASCII (American Standard Code for Information Interchange) standard is a well documented code for representing data in personal computers.

   b. Virtually all recent software supports output to an ASCII file. Database programs, word processors, spreadsheets, and even some graphics software allow such output. For those packages which do not (or where the original software is not available), there are numerous conversion programs on the market and in the public domain. Using one of these options, an ASCII output file can be generated.

   c. Reading ASCII files can be imagined as the sequential progressing through one long string of characters. Each character's location is determined by its cumulative count from the first character in the file. When you know the location of the characters in which you are interested, they can be accessed almost instantaneously– whatever the file size!

2. Isolate key terms

   a. Reducing the knowledge to study is best accomplished by selecting only those data relating to key terms of interest in the study. Standard search techniques are useful but limiting since similar terms and phrases are likely to be expressed differently by various authors. While the future may apply artificial intelligence or neural networking to the problem, an interim approach is available. Begin by conducting a raw concordance of all files to identify the unique words contained.

   b. Review the remaining words. They should represent the important words of the source. Make notes of the key terms and their variations (synonyms, abbreviations, misspellings). These will control the search and extract process later.

3. Collect information surrounding key terms

   a. Review the source files to determine the delimiters for records. This may be the standard carriage return/line feed (CR/LF) pair (normally used to delimit a single line) or a double CR/LF to denote an end of paragraph. Using a text manager such as SearchIt, Gofer, or IZE, extract the records containing the key terms into a single file. This search and extract may need to be performed iteratively for each source file and again for each key term. The end product will show one or more files containing only information relevant to the key term.

   b. The reduced data set may be placed into a data structure for storage, editing, or merge with other data, if needed. These methods can also be used to assist the analyst in finding data for a given data structure.

4. Begin content analysis

   a. The end product files should represent a substantial reduction in size from the source files.

   b. This reduced set is further organized into topics (key terms) ready for the analyst to begin the process of content analysis as applicable to the training study being conducted.

Present Application

While the prototypical file scrutiny process was being developed using APL*PLUS, it soon became evident that a compiled language would speed the looping within a program that was required by string searching each record. Parallel development of record defining, concordance, and searching was begun using the Turbo Pascal language. The result was called, "Method of ASCII Search and Text Record Extraction" (MASTRE).

MASTRE has been used on two projects in the Department of Defense during the past year. The first was a project to determine the frequency of occurrence of certain key words in task statements so that a taxonomy of tasks could be developed. The second involved identifying key words in Marine Corps Individual Training Standards that could be used for content analysis and consideration of the content of specific occupations.

Example

We begin with a trainer's problem of updating an Army program of instruction for a course teaching the Indirect Fire Infantryman, MOS11C. The course proponent already has DOS files containing all of the task numbers and titles comprising the MOS (here called MOS11C.TSK). The trainer also has a file containing equipment data (the second floppy disk in the diagram, called CROSSWLK.FTM). Figure 1 gives an overview of the process to be illustrated. Records on files from these two very different disks will now be related.

The trainer first conducts a concordance of task file MOS11C.TSK The concordance function counts all of the words in the file and reports their frequency of occurrence. The key words that may contribute to a lesson concerned with "mortars" are determined here. A temporary file, MOS11C.CNC, is developed, part of which is represented in Figure 2.

MASTRE reported that there were distinct 2378 words in this file, but some occurred more than once. There were 322 unique words summarized in this concordance file, here called MOS11C.CNC. Figure 2 shows part of the file, scrolled by the user so that the keyword "mortar" was at the center row of the screen. There were 95 appearances of this keyword. (There were also 1 each of "mortars" and the misspelled synonym, "motar,"—illustrating our earlier point about working through the misuse of language).

The training developer next uses the MASTRE search function to find the records having the term, "mortar" in the source file, MOS11C.TSK. Figure 3 reveals the first records, numbers 22, 29, ...48 of the 95 records having the term present. Record number 22 of task 11C 071-321-0505 is at the top of this report. The trainer may then scroll down to see the other records in this results file, looking for mention of equipment that should be later similarly searched in the equipment file.

Following this we examine its originating document, the MOS11C.TXT file itself. It has been scrolled to look at one of the task statements depicted in Figure 4. Record number 1 is at the top; record 22 is seen by scrolling down three lines.

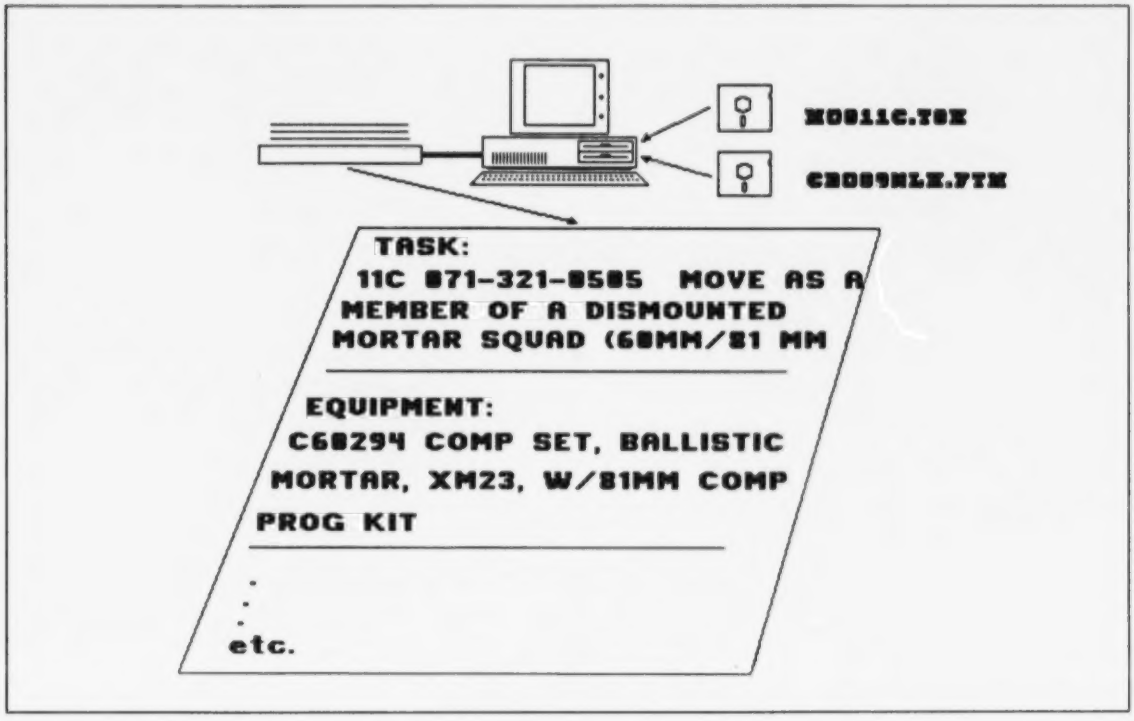

Figure 1: Relating Disparate Files

```
LIST     167              02-15-89 11:35 ♦ MOS11C.CNC
    6  met
    2  metallic
    2  meteorological
    1  method
    7  mine
    1  minefield
    4  mines
   10  misfire
   10  mission
   68  mm
    2  mode
   95  mortar
    1  mortars
    1  motar
    6  mount
   23  mounted
    1  mout
    2  mouth
    4  move
    1  movement
    1  moving
    1  navigate
    1  negotiate
Command►                                Toggles: hBkMpswTclj Keys: X=exit ?=Help
```

Figure 2: Concordance of MOS11C.TSK File

```
LIST       6     220     02-15-89 11:38 ♦ MOS11C.SRC
   22>  11C 071-321-0505    MOVE AS A MEMBER OF A DISMOUNTED MORTAR SQUAD (60MM/
81MM MORTAR)
   29>  11C 071-321-1607    RECORD INFORMATION ON FIRING DATA SHEET (81MM MORTAR
)
   31>  11C 071-321-1621    RECORD METEOROLOGICAL (MET) DATA USING MET DATA SHEE
T (81MM MORTAR)
   32>  11C 071-321-1622    DETERMINE AND APPLY MET FIRING CORRECTIONS (81MM MOR
TAR)
   40>  11C 071-321-3901    PLACE A GROUND-MOUNTED 81MM MORTAR INTO ACTION
   41>  11C 071-321-3902    BORESIGHT AN 81MM MORTAR
   42>  11C 071-321-3903    PERFORM SAFETY CHECKS ON AN 81MM MORTAR (GROUND OR C
ARRIER-MOUNTED)
   43>  11C 071-321-3904    LAY AN 81MM MORTAR FOR DEFLECTION AND ELEVATION (D&E
) (GROUND OR
   44>  11C 071-321-3905    PREPARE 81MM MORTAR AMMUNITION FOR FIRING
   45>  11C 071-321-3906    PERFORM OPERATOR MAINTENANCE ON AN 81MM MORTAR AND A
SSOCIATED FIRE
   46>  11C 071-321-3907    REMOVE A MISFIRE FROM AN 81MM MORTAR (GROUND-MOUNTED
)
   47>  11C 071-321-3909    REFER SIGHT AND REALINE AIMING POSTS FOR AN 81MM MOR
TAR
   48>  11C 071-321-3910    RECIPROCALLY LAY AN 81MM MORTAR USING AN M2 AIMING C
IRCLE AND PLACE
Command►                              Toggles: hBkMpsWTclj Keys: X=exit ?=Help
```

Figure 3: Search of MOS11C.TSK File Having "mortar"

```
LIST      1            08-04-88 10:54 ♦ MOS11C.TSK
11C 051-192-1002    INSTALL THE M16A1 BOUNDING ANTIPERSONNEL MINE (WITH/WITHOUT
TRIPWIRES)
11C 051-192-1008    INSTALL THE M21 METALLIC ANTITANK (AT) MINE
11C 051-192-1012    DISARM THE M16A1 BOUNDING ANTIPERSONNEL MINE EQUIPPED WITH/W
ITHOUT
11C 051-192-1018    DISARM THE M21 METALLIC ANTITANK (AT) MINE
11C 051-192-1021    LOCATE MINES BY VISUAL MEANS
11C 051-192-1022    LOCATE MINES BY PROBING
11C 051-192-1501    NEUTRALIZE MINES
11C 061-283-6002    LOCATE A TARGET BY SHIFT FROM A KNOWN POINT
11C 061-283-6003    CALL FOR/ADJUST INDIRECT FIRE
11C 061-283-6004    LOCATE A TARGET BY POLAR PLOT
11C 071-311-2006    USE LIMITED VISIBILITY FIRING TECHNIQUES WITH THE M16A1 RIFL
E
11C 071-311-2101    PERFORM OPERATOR MAINTENANCE ON M203 GRENADE LAUNCHER AND AM
MUNITION
11C 071-311-2103    ZERO AN M203 GRENADE LAUNCHER
11C 071-311-2105    USE LIMITED VISIBILITY FIRING TECHNIQUE WITH THE M203 GRENAD
E LAUNCHER
11C 071-313-3451    PERFORM OPERATOR MAINTENANCE ON A CALIBER .50 M2 HB MACHINEG
UN AND
11C 071-313-3452    TARGET/ZERO A CALIBER .50 MACHINEGUN
11C 071-313-3453    LOAD, REDUCE A STOPPAGE, UNLOAD AND CLEAR A CALIBER .50 MACH
Command>    *** Top of file ***        Toggles: h8kMpsWTclj Keys: X=exit ?=Help
```

Figure 4: Original MOS11C.TSK File

```
LIST      2            02-15-89 11:52 ♦ CROSSWLK.SRC
 Criteria for searching file C:\TASKS\ARMY\CROSSWLK.FTM
 => "mortar"

Line#>  Record
  154>  C60294                          COMP SET, BALLISTIC MORTAR, XM23 W/120MM
 COMP PROG KIT  11C INDIRECT FIRE INFANTRYMAN
  155>  C60294                          COMP SET, BALLISTIC MORTAR, XM23 W/120MM
 COMP PROG KIT  34L FIELD ARTILLERY DIGITAL SYSTEM REPAIRER
  189>  D10741                          CARRIER 107 MILLIMETER MORTAR: SELF PROP
ELL             29E
  190>  D10741                          CARRIER 107 MILLIMETER MORTAR: SELF PROP
ELL             31V
  191>  D10741                          CARRIER 107 MILLIMETER MORTAR: SELF PROP
ELL             41C
  192>  D10741                          CARRIER 107 MILLIMETER MORTAR: SELF PROP
ELL             63H
  193>  D10741                          CARRIER 107 MILLIMETER MORTAR: SELF PROP
ELL             63J
  194>  D10741                          CARRIER 107 MILLIMETER MORTAR: SELF PROP
ELL             63Y
  195>  D10741                          CARRIER 107 MILLIMETER MORTAR: SELF PROP
ELL             76Y
  196>  D10741                          CARRIER 107 MILLIMETER MORTAR: SELF PROP
Command>                               Toggles: h8kMpsWTclj Keys: X=exit ?=Help
```

Figure 5: Search of CROSSWLK.FTM for "mortar"

The equipment file, an ASCII data file called CROSSWLK.FTM, was downloaded from the mainframe FOCUS Equipment-to-Occupation CrossWalk Data Base maintained by the Defense Training and Performance Data Center. Calling the same search function used on the .TSK file we again search the CROSSWLK.FTM file for "mortar." The result is presented in Figure 5. Any record having the term "mortar" present is thus revealed. This particular file shows a number, Army military occupational specialty, and a description of the equipment associated with the term.

The example has shown how specific information was located in two unrelated text files with different languages and different computers. They revealed specific information that a training developer may use for the development of a program of instructions pertaining to mortars in the 11C infantry military occupational specialty.

Discussion

Text can be stored primarily two ways in a computer: (1) in ASCII flat files that sequentially place all the characters in a document, including the CR/LFs and print characteristics that give the document its shape when recalled with something like the DOS print function; and (2) in structured files that are really flat files with the location (in number of bytes from the origin of the file) of records, data elements, or other structure imposed by a data base language.

In the former, the viewer scrolls through the "bit stream" sequentially using the default wordwrapping of the computer read program. While the DOS "type" command only moves one way, there are other read programs that allow scrolling upward, downward, or to points the reader may wish to define. (Vernon Buerg's "List" shareware utility was used to present the information in Figures 2-5.) Little planning is needed to do this generally linear progression through files.

In the latter, some systems analysis is required to define what use you want for the computer system ahead of time. The data base manager must define the entities, data elements, interrelationships, and identify the location of each in this structure. Later changes in the plan require restructuring of the files.

The procedures delineated here provide an alternative to the predefined data base approach to locating data in a computer system. Each has its advantages and disadvantages. While the data base management approach is superior to the rather unplanned, desultory procedures allowed by string searching ASCII files for relationships that were not foreseen at the time of data base design, this superior approach is not always possible. In such cases, the thesis of this paper suggests that all is not lost. Records can be found in flat files.

Future Application

Better communication is the outcome. Future development will have a utility program enabling a non-technical person to determine the native language and file structure of a given file, read and edit that file in whatever language is friendly for the user, and present output in any native language and file type desired.

This notion is similar to that presented for the word processing realm in Borland's "Sprint." Sprint has a dozen or so user interfaces and import/export capabilities. For example, users can read a document printed in Word Perfect, interact with its file in a WordStar interface, and produce a MultiMate document file. The data base world is close to having the same capabilities.

Artificial intelligence may have some utility in defining key words, essential meaning, synonmys, antonyms, and misspelled words. Future development will surely include data reduction procedures; utilities for determining the essential meaning of a paragraph, perhaps in the way a determinant can describe a matrix. Neural networking procedures seem to have some potential for this problem.